# LAPORAN PROYEK MENAMPILKAN SUHU, KELEMBAPAN DAN INTENSITAS CAHAYA PADA OLED

Disusun untuk memenuhi tugas mata kuliah Internet of Things

Dosen Pengampu:

Ir. Subairi, ST., MT., IPM



Disusun Oleh:

Mochammad Risqullah Adiyananta

233140707111099

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**DEPARTEMEN INDUSTRI KREATIF DAN DIGITAL**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2025**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc192667757)

[ABSTRAK 4](#_Toc192667758)

[BAB I 5](#_Toc192667759)

[PENDAHULUAN 5](#_Toc192667760)

[1.1 Latar Belakang 5](#_Toc192667761)

[1.2 Tujuan Praktikum 5](#_Toc192667762)

[BAB II 6](#_Toc192667763)

[METODOLOGI 6](#_Toc192667764)

[2.1 Alat dan Bahan 6](#_Toc192667765)

[2.2 Langkah Implementasi 6](#_Toc192667766)

[BAB III 7](#_Toc192667767)

[HASIL DAN PEMBAHASAN 7](#_Toc192667768)

[3.1 Hasil Eksperimen 7](#_Toc192667769)

[3.2 Kode Program 7](#_Toc192667770)

[3.2.1 Main.cpp 7](#_Toc192667771)

[3.2.2 Diagram.json 10](#_Toc192667772)

[BAB IV 11](#_Toc192667774)

[KESIMPULAN 11](#_Toc192667775)

[4.1 Kesimpulan 11](#_Toc192667776)

# ABSTRAK

Pada eksperimen ini, sistem berbasis ESP32 dirancang untuk memantau suhu, kelembapan, dan intensitas cahaya menggunakan sensor DHT22, LDR, dan layar OLED. Tujuan dari penelitian ini adalah untuk mengimplementasikan sistem pemantauan real-time yang dapat menampilkan data lingkungan secara langsung. Sensor DHT22 digunakan untuk mengukur suhu dan kelembapan, sementara LDR digunakan untuk mendeteksi intensitas cahaya. Data yang diperoleh dari kedua sensor tersebut ditampilkan secara real-time pada layar OLED. Hasil simulasi menunjukkan bahwa sistem dapat secara efektif mengukur dan menampilkan suhu, kelembapan, serta intensitas cahaya dengan akurasi yang baik. Pembacaan data diperbarui secara berkala dalam interval waktu tertentu, menghasilkan pemantauan yang konsisten dan stabil. Implementasi ini menunjukkan bahwa ESP32 bersama dengan sensor DHT22 dan LDR dapat digunakan sebagai solusi pemantauan lingkungan yang sederhana namun efektif.

*Kata kunci: ESP32, sensor DHT22, LDR, layar OLED, pemantauan real-time, suhu, kelembapan, intensitas cahaya.*

In this experiment, an ESP32-based system was designed to monitor temperature, humidity, and light intensity using the DHT22 sensor, LDR, and OLED display. The aim of this research was to implement a real-time monitoring system that can display environmental data directly. The DHT22 sensor was used to measure temperature and humidity, while the LDR detected light intensity. The data collected from both sensors was displayed in real-time on the OLED screen. The simulation results demonstrated that the system effectively measured and displayed temperature, humidity, and light intensity with good accuracy. Data readings were updated periodically at specific intervals, resulting in consistent and stable monitoring. This implementation shows that ESP32, together with the DHT22 and LDR sensors, can serve as a simple yet effective environmental monitoring solution.

*Keyword* *ESP32, DHT22 sensor, LDR, OLED display, real-time monitoring, temperature, humidity, light intensity.*

# BAB I

# PENDAHULUAN

## Latar Belakang

Perkembangan teknologi berbasis IoT (Internet of Things) semakin pesat, memudahkan berbagai pengaplikasian sensor dalam kehidupan sehari-hari, termasuk pemantauan kondisi lingkungan seperti suhu, kelembapan, dan intensitas cahaya. Dalam proyek ini, digunakan ESP32 sebagai mikrokontroler, DHT22 sebagai sensor suhu dan kelembapan, LDR (Light Dependent Resistor) untuk mendeteksi intensitas cahaya, dan OLED sebagai display untuk menampilkan data. Penerapan sensor-sensor ini membantu dalam pengumpulan data lingkungan secara real-time.

## Tujuan Praktikum

1. Mengukur dan menampilkan suhu dan kelembapan lingkungan menggunakan sensor DHT22.
2. Mengukur intensitas cahaya menggunakan LDR.
3. Menampilkan hasil pengukuran pada layar OLED secara real-time.

# BAB II

# METODOLOGI

## Alat dan Bahan

1. ESP32
2. Sensor DHT22
3. LDR
4. Layar OLED 128x64
5. Software Arduino IDE/PlatformIO
6. Wokwi simulator

## Langkah Implementasi

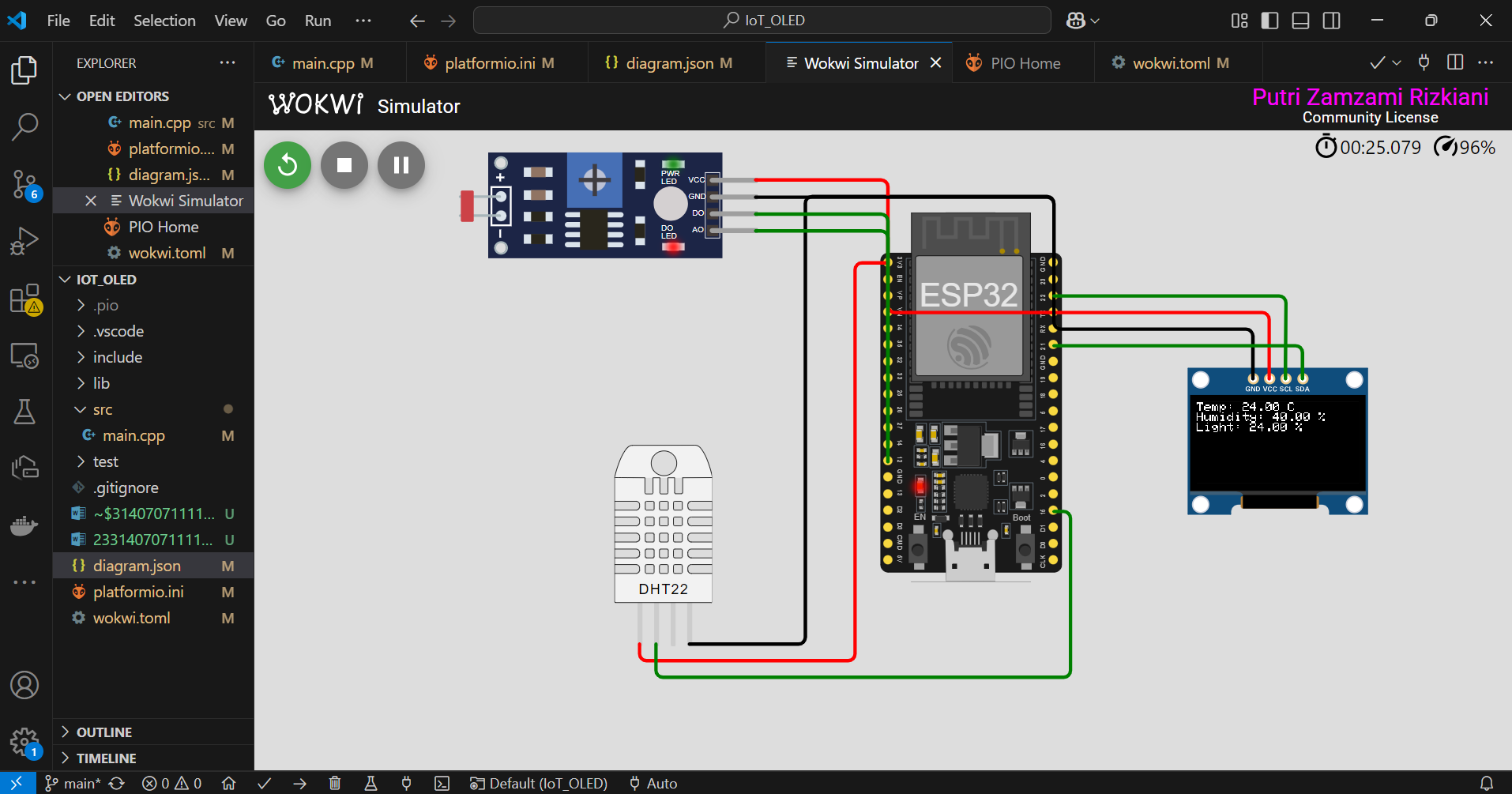
1. Menghubungkan ESP32 dengan sensor DHT22, LDR, dan OLED sesuai dengan skematik yang dirancang
2. Menggunakan library untuk OLED dan DHT22 di Arduino IDE atau PlatformIO
3. Inisialisasi sensor dan tampilkan hasil pengukuran suhu, kelembapan, dan intensitas cahaya ke OLED secara berkala.
4. Melakukan pengujian dan simulasi. Setelah program diunggah ke ESP32, hasil pembacaan sensor akan muncul di layar OLED. Simulasi bisa dilakukan terlebih dahulu menggunakan platform Wokwi.

# BAB III

# HASIL DAN PEMBAHASAN

## Hasil Eksperimen

Berdasarkan hasil praktikum, ini menunjukkan bahwa ESP32 mampu menerima data dari sensor DHT22 untuk mengukur suhu dan kelembapan, serta mengukur intensitas cahaya menggunakan LDR. Data yang diperoleh kemudian ditampilkan pada layar OLED secara real-time. Sebagai contoh, data yang diperoleh dari hasil simulasi menunjukkan suhu sebesar 24.00°C, kelembapan 40.00%, dan intensitas cahaya 24.00%. Data ini diperbarui secara berkala dengan setiap pembacaan diambil dalam interval tertentu, misalnya setiap 5 detik. Sensor DHT22 memberikan hasil yang stabil dan akurat untuk pengukuran suhu dan kelembapan, sedangkan LDR mampu mendeteksi perubahan intensitas cahaya dengan baik, memberikan pemantauan lingkungan yang efektif.



Gambar 1. (Simulasi)

## Kode Program

### 3.2.1 Main.cpp

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <Adafruit\_GFX.h>

#include <DHT.h>

#define SCREEN\_WIDTH 128 // OLED width, in pixels

#define SCREEN\_HEIGHT 64 // OLED height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// DHT22 setup

#define DHTPIN 15      // Pin where DHT22 is connected

#define DHTTYPE DHT22  // DHT 22 (AM2302)

DHT dht(DHTPIN, DHTTYPE);

// LDR setup

#define LDR\_PIN 12     // Analog pin for LDR (AO pin)

void setup() {

  // Initialize the serial monitor

  Serial.begin(115200);

  // Initialize the DHT22 sensor

  dht.begin();

  // Initialize the OLED display

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {  // Address 0x3C for 128x64

    Serial.println(F("SSD1306 allocation failed"));

    for (;;); // Don't proceed, loop forever

  }

  // Clear the display buffer

  display.clearDisplay();

  // Set text size and color

  display.setTextSize(1);

  display.setTextColor(SSD1306\_WHITE);

}

void loop() {

  // Read temperature and humidity from DHT22

  float temperature = dht.readTemperature();

  float humidity = dht.readHumidity();

  // Read light intensity from LDR

  int lightValue = analogRead(LDR\_PIN);

  // Convert LDR value to a percentage (optional, for display)

  float lightPercentage = map(lightValue, 0, 4095, 0, 100);

  // Check if any reading failed, and exit early (to try again).

  if (isnan(temperature) || isnan(humidity)) {

    Serial.println(F("Failed to read from DHT sensor!"));

    return;

  }

  // Print the values to the serial monitor (optional)

  Serial.print(F("Temperature: "));

  Serial.print(temperature);

  Serial.print(F(" °C  Humidity: "));

  Serial.print(humidity);

  Serial.print(F("%  Light Intensity: "));

  Serial.print(lightPercentage);

  Serial.println(F("%"));

  // Display the values on OLED

  display.clearDisplay();  // Clear the display buffer

  display.setCursor(0, 0); // Set the cursor to (0,0)

  display.print(F("Temp: "));

  display.print(temperature);

  display.println(F(" C"));

  display.print(F("Humidity: "));

  display.print(humidity);

  display.println(F(" %"));

  display.print(F("Light: "));

  display.print(lightPercentage);

  display.println(F(" %"));

  display.display();  // Send the buffer to the display

  delay(2000);  // Wait for 2 seconds before the next reading

}

# 3.2.2 Diagram.json

# 

# BAB IV

# KESIMPULAN

## Kesimpulan

Proyek ini berhasil melakukan pengukuran suhu, kelembapan, dan intensitas cahaya menggunakan sensor DHT22, LDR, dan ESP32. Data yang diperoleh ditampilkan secara real-time pada layar OLED. Penggunaan ESP32 sebagai mikrokontroler sangat tepat karena mendukung komunikasi dengan berbagai sensor dan memiliki kapasitas yang cukup untuk menjalankan program yang kompleks. Proyek ini bisa dikembangkan lebih lanjut dengan penambahan fitur seperti penyimpanan data secara online menggunakan platform IoT.